import openpyxl

from datetime import datetime

import re

# Verification function for student ID format

def validate\_id\_of\_Student (input\_id):

# Alphanumeric characters are matched by a regular expression pattern

pattern = r'^[A-Za-z0-9]+$'

return bool(re.match(pattern, input\_id))

# Validation function for semester date format

def validate\_dates\_of\_Semester(input\_date):

try:

datetime.strptime(input\_date, "%m/%d/%Y")

except ValueError:

try:

datetime.strptime(input\_date, "%d-%m-%Y")

except ValueError:

return False

return True

#Function to verify the student's CGPA format

def validate\_CGPA\_of\_Student(input\_cgpa):

if input\_cgpa.upper() == 'N/A':

return True

pattern = r'^\d+(\.\d{1,2})?$'

return bool(re.match(pattern, input\_cgpa))

# Function to determine a student's row number by name

def find\_studentname(sheet, name):

for i in range(2, sheet.max\_row + 1):

if sheet.cell(row=i, column=1).value == name:

return i

return None

# Capability to show every student record

def display\_students\_records(sheet):

print('\nCurrent Student Records:\n')

print(f'{"Name":<20}{"ID":<10}{"Start Date":<15}{"End Date":<15}{"CGPA":<10}{"Course":<20}{"Duration":<15}')

for i in range(2, sheet.max\_row + 1):

# Get values from the current row's cells.

name = sheet.cell(row=i, column=1).value

student\_id = sheet.cell(row=i, column=2).value

start\_date = sheet.cell(row=i, column=3).value

end\_date = sheet.cell(row=i, column=4).value

cgpa = sheet.cell(row=i, column=5).value

course = sheet.cell(row=i, column=6).value

# Use the calculate\_semester\_duration function to determine the duration.

duration = calculate\_semester\_duration(start\_date, end\_date)

# Publish information from student records in format.

print(f'{name:<20}{student\_id:<10}{start\_date:<15}{end\_date:<15}{cgpa:<10}{course:<20}{duration:<15}')

# Function to determine the length of a semester

def calculate\_semester\_duration(start\_date, end\_date):

if start\_date is None or end\_date is None:

raise ValueError("Start date and end date cannot be None.")

# Experiment with parsing the start\_date in various date formats.

try:

start\_date\_obj = datetime.strptime(start\_date, "%m/%d/%Y")

except ValueError:

try:

start\_date\_obj = datetime.strptime(start\_date, "%d-%m-%Y")

except ValueError:

raise ValueError("Invalid start date format. Please use MM/DD/YYYY or DD-MM-YYYY.")

# Experiment with parsing the end\_date in various date formats.

try:

end\_date\_obj = datetime.strptime(end\_date, "%m/%d/%Y")

except ValueError:

try:

end\_date\_obj = datetime.strptime(end\_date, "%d-%m-%Y")

except ValueError:

raise ValueError("Invalid end date format. Please use MM/DD/YYYY or DD-MM-YYYY.")

duration = end\_date\_obj - start\_date\_obj

return f'{duration.days} days'

# Password authentication function

def Password\_function():

correct\_password = "7070"

attempts\_left = 4

while attempts\_left > 0:

user\_password = input("Enter password: ")

if user\_password == correct\_password:

return True

else:

attempts\_left -= 1

print(f'Incorrect password. Attempts left: {attempts\_left}')

print("Authentication failed. Exiting program.")

return False

def main():

# File path for the Excel workbook

workbook\_path = r'D:\Semester 2\Document Automation Python\Final Project\student\_Record\_File.xlsx'

workbook = openpyxl.load\_workbook(workbook\_path)

worksheet = workbook.active

if not Password\_function():

exit()

print('\n=== Welcome to the Custom Student Records System ===')

while True:

print('\nOptions:')

print('1. Add a new student record')

print('2. Update an existing student record')

print('3. Delete a student record')

print('4. View the list of student records')

print('5. Search and display a specific student record')

print('6. Exit')

choice\_of\_user= input('Enter your choice you want to do: ')

# Option to add a new student record

if choice\_of\_user== '1':

student\_name = input('Enter the name of the student: ')

row = find\_studentname(worksheet, student\_name)

if row is not None:

print('Student record already exists.')

else:

#Ask the user for additional information if the student record is missing.

student\_id = input('Enter the student ID: ')

while not validate\_id\_of\_Student (student\_id):

student\_id = input(int('Invalid input. Enter the student ID: '))

start\_date = input('Enter the start date of the semester (MM/DD/YYYY or DD-MM-YYYY): ')

while not validate\_dates\_of\_Semester (start\_date):

start\_date = input('Invalid input. Enter the start date of the semester (MM/DD/YYYY or DD-MM-YYYY): ')

end\_date = input('Enter the end date of the semester (MM/DD/YYYY or DD-MM-YYYY): ')

while not validate\_dates\_of\_Semester (end\_date):

end\_date = input('Invalid input. Enter the end date of the semester (MM/DD/YYYY or DD-MM-YYYY): ')

input\_cgpa = input('Enter the CGPA of the student (or N/A if not applicable): ')

while not validate\_CGPA\_of\_Student(input\_cgpa):

input\_cgpa = input('Invalid input. Enter the CGPA of the student (or N/A if not applicable): ')

cgpa = 'N/A' if input\_cgpa.upper() == 'N/A' else f'{float(input\_cgpa):.2f}%'

course = input('Enter the course of the student: ').upper()

worksheet.append([student\_name, student\_id, start\_date, end\_date, cgpa, course])

workbook.save(workbook\_path)

print('Student record added successfully!')

# Option to update an existing student record

elif choice\_of\_user== '2':

student\_name = input('Enter the name of the student to update: ')

row = find\_studentname(worksheet, student\_name)

if row is not None:

# Show the specifics of the discovered student record.

print(f"\nStudent Record Found:\nName: {worksheet.cell(row=row, column=1).value}\nID: {worksheet.cell(row=row, column=2).value}\nStart Date: {worksheet.cell(row=row, column=3).value}\nEnd Date: {worksheet.cell(row=row, column=4).value}\nCGPA: {worksheet.cell(row=row, column=5).value}\nCourse: {worksheet.cell(row=row, column=6).value}")

new\_id = input('Enter the new student ID: ')

while not validate\_id\_of\_Student (new\_id):

new\_id = input('Invalid input. Enter the new student ID: ')

new\_start\_date = input('Enter the new start date of the semester (MM/DD/YYYY or DD-MM-YYYY): ')

while not validate\_dates\_of\_Semester(new\_start\_date):

new\_start\_date = input('Invalid input. Enter the new start date of the semester (MM/DD/YYYY or DD-MM-YYYY): ')

new\_end\_date = input('Enter the new end date of the semester (MM/DD/YYYY or DD-MM-YYYY): ')

while not validate\_dates\_of\_Semester(new\_end\_date):

new\_end\_date = input('Invalid input. Enter the new end date of the semester (MM/DD/YYYY or DD-MM-YYYY): ')

new\_input\_cgpa = input('Enter the new CGPA of the student (or N/A if not applicable): ')

while not validate\_CGPA\_of\_Student(new\_input\_cgpa):

new\_input\_cgpa = input('Invalid input. Enter the new CGPA of the student (or N/A if not applicable): ')

new\_cgpa = 'N/A' if new\_input\_cgpa.upper() == 'N/A' else f'{float(new\_input\_cgpa):.2f}%'

new\_course = input('Enter the new course of the student: ').upper()

# Add the new information for the discovered student record to the worksheet.

worksheet.cell(row=row, column=2, value=new\_id)

worksheet.cell(row=row, column=3, value=new\_start\_date)

worksheet.cell(row=row, column=4, value=new\_end\_date)

worksheet.cell(row=row, column=5, value=new\_cgpa)

worksheet.cell(row=row, column=6, value=new\_course)

workbook.save(workbook\_path)

print('Student record updated successfully!')

else:

print('Student record not found.')

# Option to delete a student record

elif choice\_of\_user== '3':

student\_name = input('Enter the name of the student to delete: ')

row = find\_studentname(worksheet, student\_name)

if row is not None:

worksheet.delete\_rows(row, amount=1)

workbook.save(workbook\_path)

print('Student record deleted successfully!')

else:

print('Student record not found.')

# Option to view the list of student records

elif choice\_of\_user== '4':

display\_students\_records (worksheet)

# Option to search and display a specific student record

elif choice\_of\_user== '5':

print('\nSearch and display a specific student record: ')

student\_name = input('Enter the name of the student: ')

row = find\_studentname(worksheet, student\_name)

if row is not None:

print(f'\nStudent Record Found:\nName: {worksheet.cell(row=row, column=1).value}\nID: {worksheet.cell(row=row, column=2).value}\nStart Date: {worksheet.cell(row=row, column=3).value}\nEnd Date: {worksheet.cell(row=row, column=4).value}\nCGPA: {worksheet.cell(row=row, column=5).value}\nCourse: {worksheet.cell(row=row, column=6).value}')

else:

print('Student record not found.')

# Option to exit the program

elif choice\_of\_user== '6':

print('\nExiting program of Final Project...')

break

else:

print('\nInvalid choice. Please try again.')

if \_\_name\_\_ == "\_\_main\_\_":

main()